

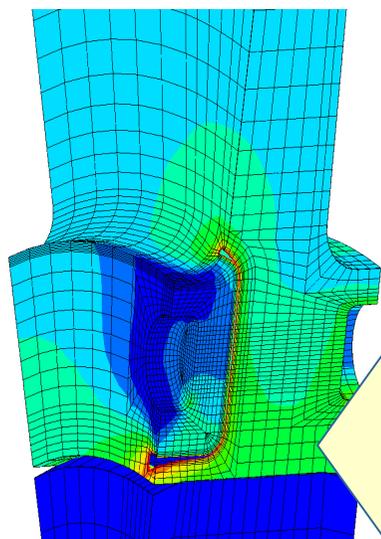
Usage de Python dans *Code_Aster*



Code_Aster et la modélisation en mécanique des structures à EDF

... en quelques mots

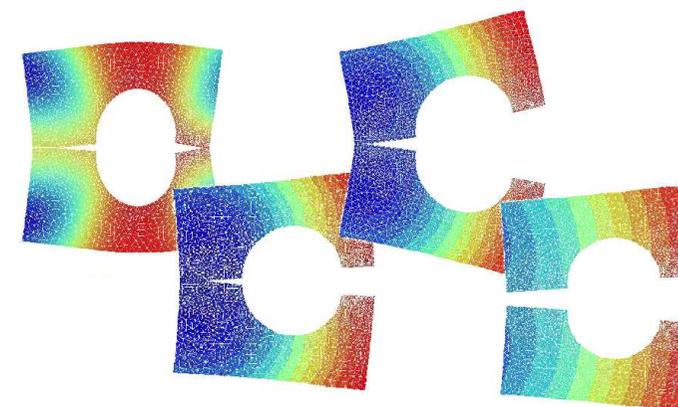
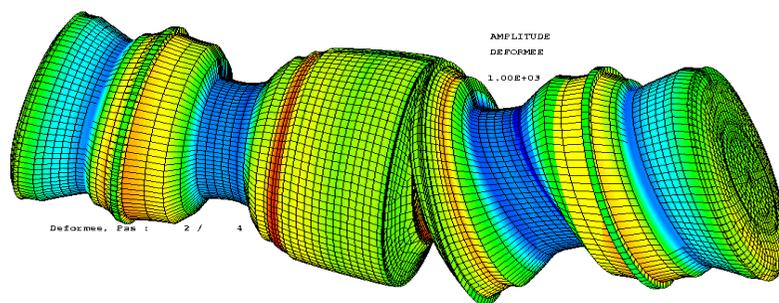
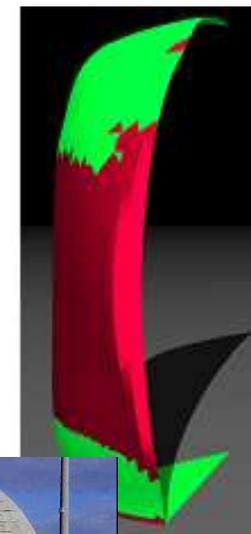
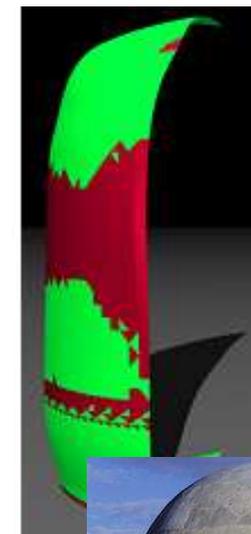
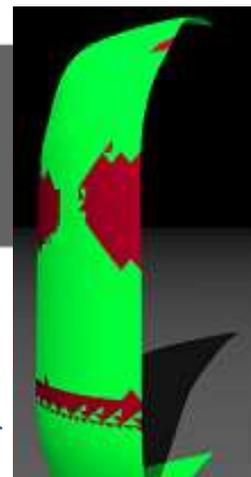
1 Un code généraliste (2/5)



↑

endommagement
rupture
fatigue...

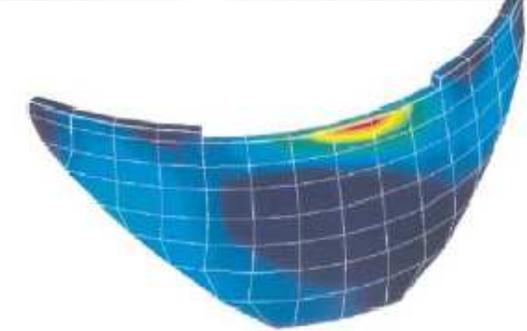
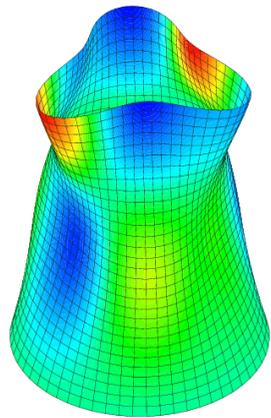
→



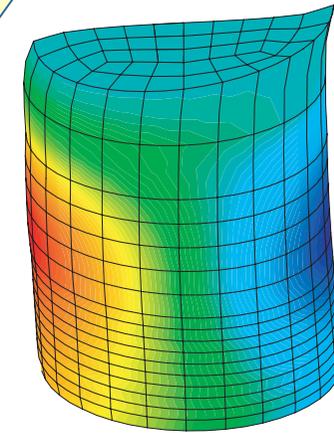
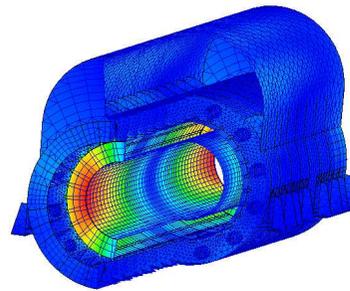
1 Un code généraliste (3/5)



dynamique
acoustique
force fluide
IFS, ISS...



FEA 7 - 30.592 Hz
Element Set 27



VAL ISO
> -5.63E+02
< 9.61E+02

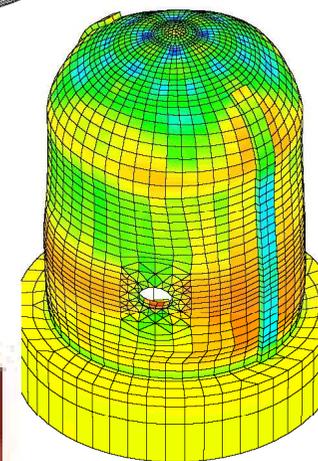
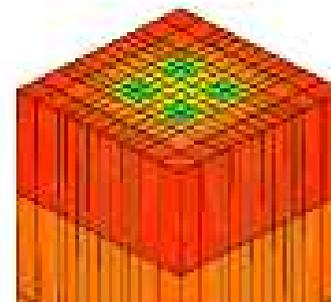
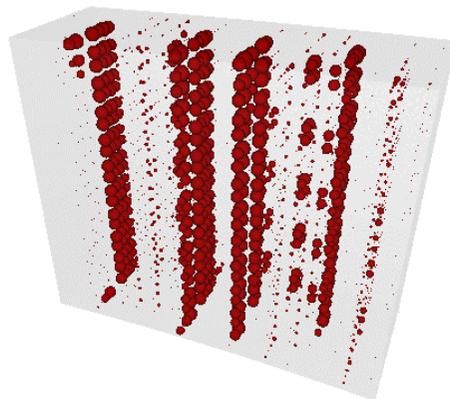
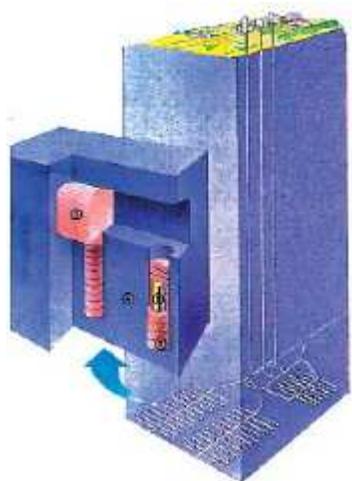
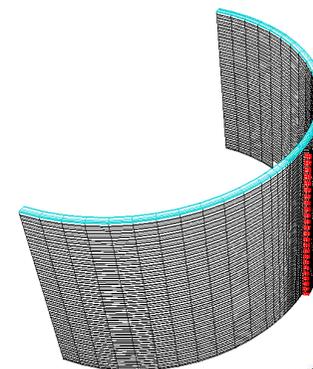
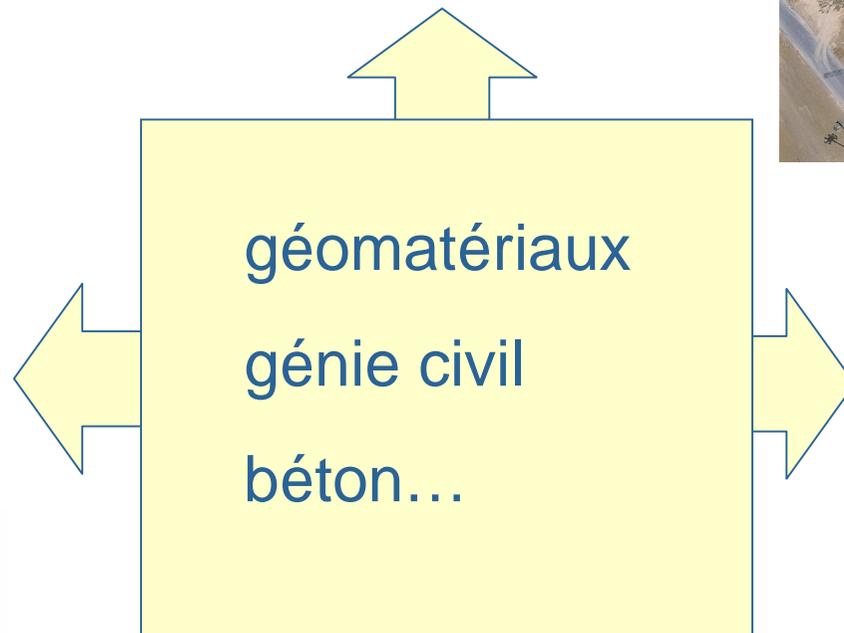
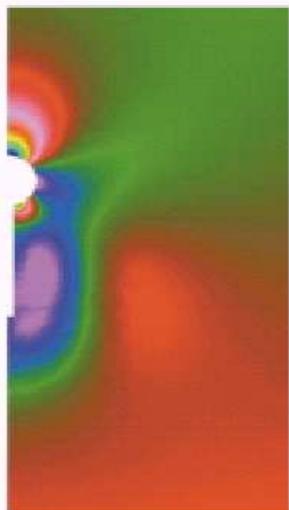
-5.51E+02
-4.79E+02
-4.08E+02
-3.36E+02
-2.65E+02
-1.94E+02
-1.22E+02
-51.
21.
92.
1.63E+02
2.35E+02
3.06E+02
3.78E+02
4.49E+02
5.20E+02
5.92E+02
6.63E+02
7.35E+02
8.06E+02
8.77E+02
9.49E+02

AMPLITUDE
DEFORMEE

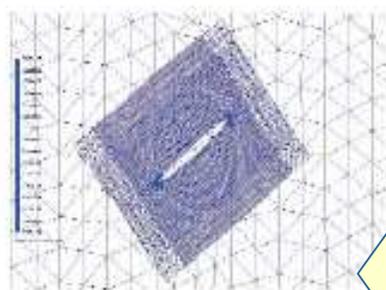
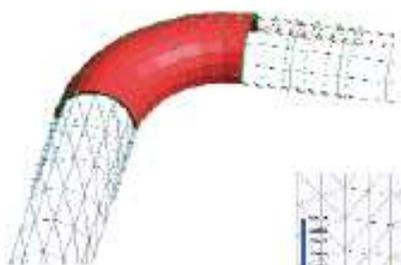
5.00E+03



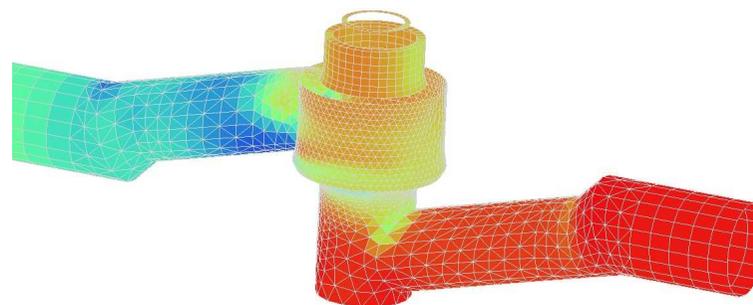
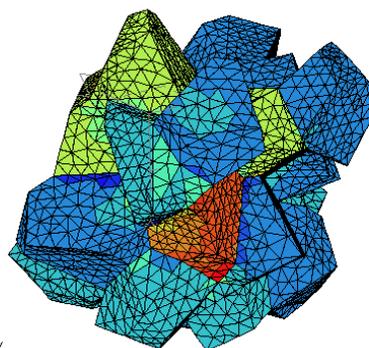
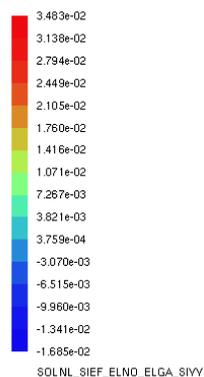
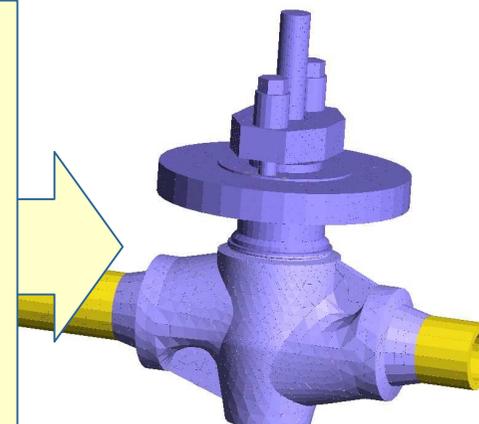
1 Un code généraliste (4/5)



1 Un code généraliste (5/5)



Multiéchelle
Multiphysique
Couplage
Chaînage...





Contexte de développement

2 Pourquoi EDF développe son propre code ?

EDF est exploitant de matériels et d'ouvrages

- Parfois co-concepteur mais pas producteur de matériels,
- Enjeux différents de ceux de l'ingénierie manufacturière.

Spécificités du domaine nucléaire

- Conception réglementaire,
- Exigences de sûreté,
- Durée de vie des ouvrages.

La place de la R&D (et de ses logiciels)

- Comprendre un événement imprévu à la conception,
- Quantifier des marges/ étude de conception,
- Justifier l'utilisation d'un matériel ou d'un process.

2 Pourquoi EDF développe son propre code ?

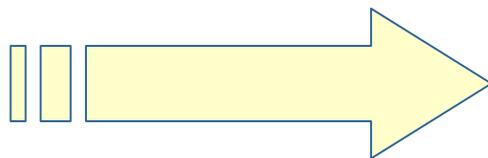
Maîtrise des **modèles numériques**

Capitalisation les travaux de **R&D** dans un code unique

Transfert rapide vers l'**ingénierie**

Indépendance vis-à-vis des éditeurs de logiciels

Développement d'outils-métiers



**Choix d'un modèle de
développement en interne**

2 L'effort de développement

Code_Aster en quelques chiffres :

- 1.300.000 lignes de code ... dont 75.000 en python
- 14.000 pages de documentation
- 2.000 cas tests

Versionnement

- Version industrielle tous les 2 ans (release semestrielle)
- Incrément de version de dvpt hebdomadaire

Organisation en réseau (type logiciel libre)

- Pas de cycle en V, pas de cahier des charges global,
- « Core team » [~20 lan] + projets métiers [~40 lan]

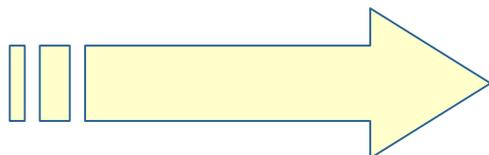
2 Un logiciel libre !

Pourquoi ?

- Accroître la qualité du code par l'usage
- Faire naître des partenariats

Quels résultats après 5 ans ?

- Une communauté active: 16 000 téléchargements, 10 000 visites/sem sur le site, forums actifs, journées Aster Libre...
- Des partenariats: ECP, ECN, IFP, ANDRA, INSA ...
- Des appropriations par les entreprises et les écoles/universités, pour lesquelles on n'a pas toujours de retour d'information...



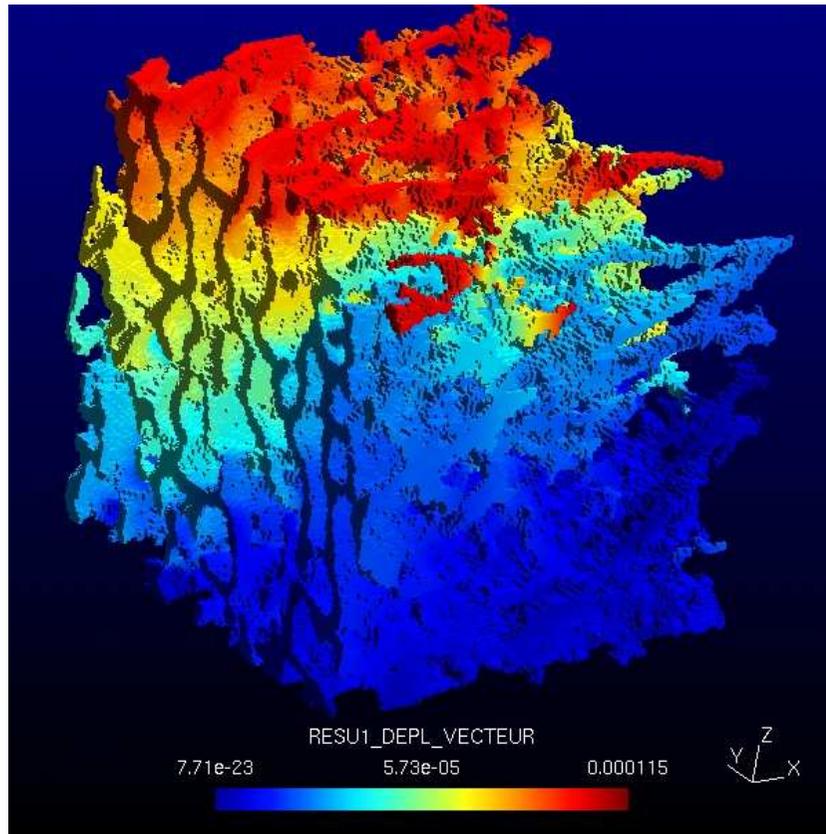
<http://www.code-aster.org>

2 Un logiciel libre !



The screenshot shows the Code_Aster website homepage. At the top left is the Code_Aster logo. The main header reads "Bienvenue sur le site de Code_Aster". To the right of the header are navigation links: RECHERCHER, PLAN / AIDE, NOUS CONTACTER, and a login prompt "Vous n'êtes pas identifié...". Below the header is a horizontal menu with tabs for PRODUIT, UTILISATION, DOCUMENTATION, DÉVELOPPEMENT, FORUM, SERVICES, TÉLÉCHARGEMENT, and IDENTIFICATION. The main content area features a "Bienvenue sur le site web de Code_Aster!" message. On the right, there are links for press releases in French, English, German, and Spanish. A news item dated 29/06/06 is highlighted, featuring a cartoon rabbit holding a star and announcing that Code_Aster has received the "Lutèce d'Or" award for "Best Free Project for a Large Group". The text explains that the award was presented at the Paris Capitale du Libre event on June 26, 2006, and lists the objectives of the award: to highlight developers, companies, and institutions; to recognize their work; and to inspire and stimulate new initiatives. Below this, two other news items are listed: one dated 29/06/06 about Code_Aster Libre version 8.3 (sources) being available, and another dated 07/01/06 about Code_Aster Libre version 8.2 (sources) being available. A final news item dated 27/10/05 mentions the Code_Aster Libre day on Civil Engineering held by EDF R&D and NECS on October 27, 2005. On the left side of the page, there is a "Plaque de présentation de la V7" with a "NEW" tag and a link to the "English version". Below this is a thumbnail image of the Code_Aster software box, which features a colorful abstract design and the text "Code_Aster Analysis of Structures and Thermomechanics for Surveys and Research". Logos for NECS and EDF are visible at the bottom of the thumbnail.

2 Un logiciel libre !



Post sur forum 08/12/06 :

Université de Sherbrooke
Os de fémur
3 M de DDL



Pourquoi Python ? Le besoin historique

3 – Un code structuré par son langage de commandes

```
# Exemple de fichier de commandes Aster  
  
DEBUT( )  
  
ma=LIRE_MALLAGE( )  
  
mat=DEFI_MATERIAU(ELAS=_F(      NU = 0.3,  
                               E = 2.E5, ), )
```

Nom utilisateur du concept produit de la commande

Mot clé facteur

```
chmat = AFFE_MATERIAU ( AFFE =_F( MATER = mat,  
                               TOUT = 'OUI', ),  
.....  
MAILLAGE = ma, )  
  
FIN( )
```

Nom de commande

Mots clés simples

3 – Un code structuré par son langage de commandes

```
chmat = AFFE_MATERIAU ( AFFE =_F( MATER = mat ,  
                          TOUT = 'OUI' , ) ,  
                          MAILLAGE = ma , )
```

- Un langage riche pour un code
- généraliste
 - multi-physique
 - complet
 - industriel (métier)
 - sans pré-post intégré
- 213 **commandes**
 - environ 9.000 **mots clés**
 - 106 **structures de données**

3 – Un code structuré par son langage de commandes

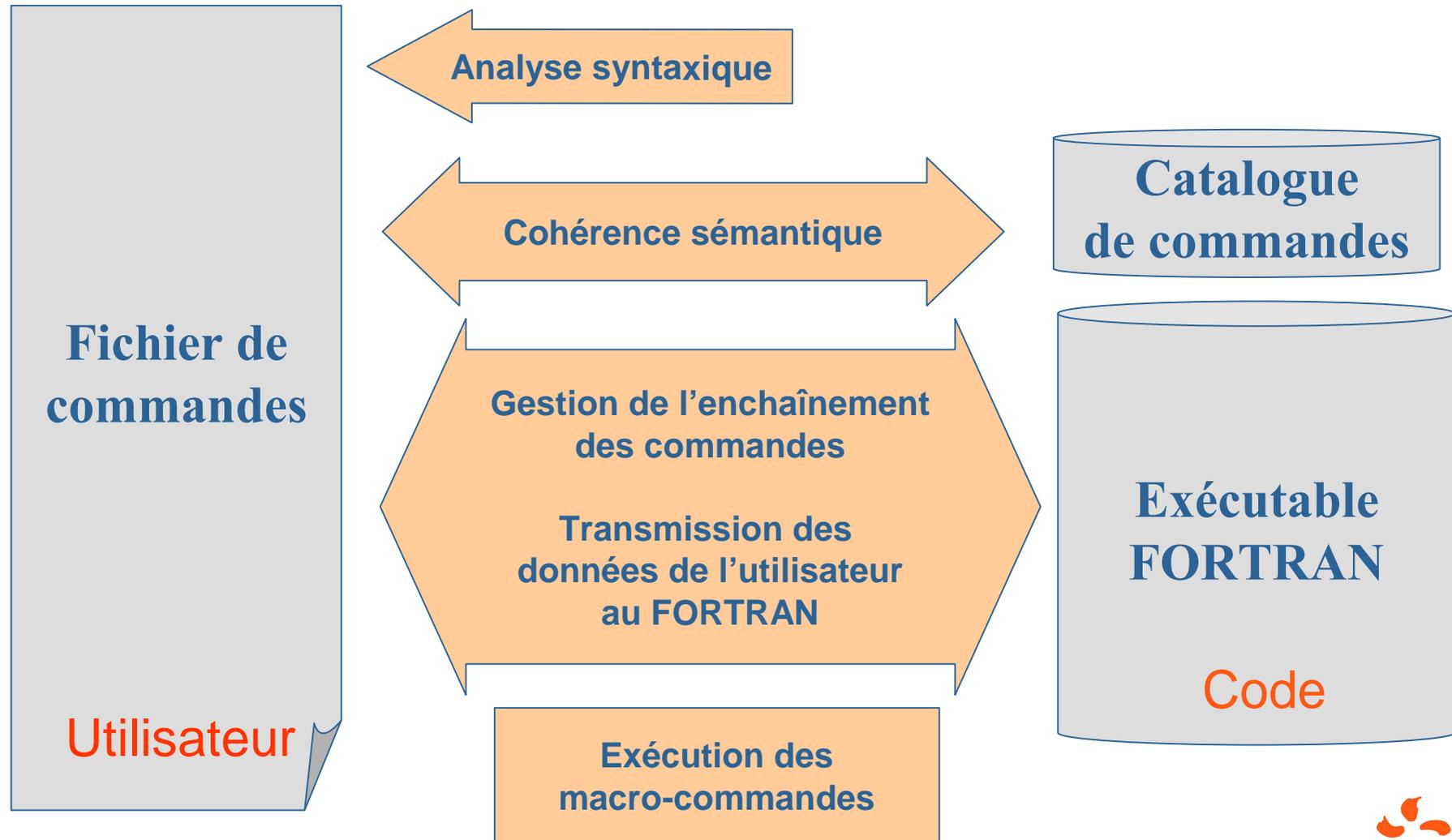
```
DEBUT(); # Start of script
PRE_GIBI(); # Prepare for reading a GIBI file
MAIL=LIRE_MAILLAGE(); # Read the mesh and call it MAIL
# define a mechanical model to use on each parts of the mesh
MODMECA=AFPE_MODELE(MAILLAGE=MAIL,
                    AFPE=( _F(GROUP_MA=('TUY1', 'TUY2',),
                              PHENOMENE='MECANIQUE',
                              MODELISATION='POU_D_T',),
                          _F(GROUP_MA='COUDE',
                              PHENOMENE='MECANIQUE',
                              MODELISATION='POU_C_T',),),),);

# Set boundary conditions
BLOQUAGE=AFPE_CHAR_MECA(MODELE=MODMECA,
                        DDL_IMPO=_F(GROUP_NO='PA',
                                      DX=0.0,
                                      DY=0.0,
                                      DZ=0.0,
                                      DRX=0.0,
                                      DRY=0.0,
                                      DRZ=0.0,),),);

CHARGT=AFPE_CHAR_MECA(MODELE=MODMECA,
                       FORCE_NODALE=_F(GROUP_NO='PB',
                                         FY=100000.0,),),);

# Solve using a static mechanic solver
RESU1=MECA_STATIQUE(MODELE=MODMECA,
                    CHAM_MATER=CHMAT,
                    CARA_ELEM=CARA_POU,
                    EXCIT=( _F(CHARGE=BLOQUAGE, ),
```

3 – Le besoin historique : supervision des calculs



3 L'histoire du choix de python pour Aster

1996-1999 : Aster-expert

- Enrichir la solution « solveur de mécanique » Aster d'une interface de mise en données et d'une plate-forme de modelling/maillage/post
- Mise en données (EFICAS) : solution C++ et LISP
- Plate-forme de modélisation : Euclid-analyst et bibliothèque CAS.Cade déclinée sur Aster et Samcef-design

Abandon en 1999 (difficultés contractuelles)

- Proposition de ré-écriture du couple superviseur/catalogues
- Partage avec EFICAS, éditeur de commandes
- Utilisation d'un langage existant (simplification)

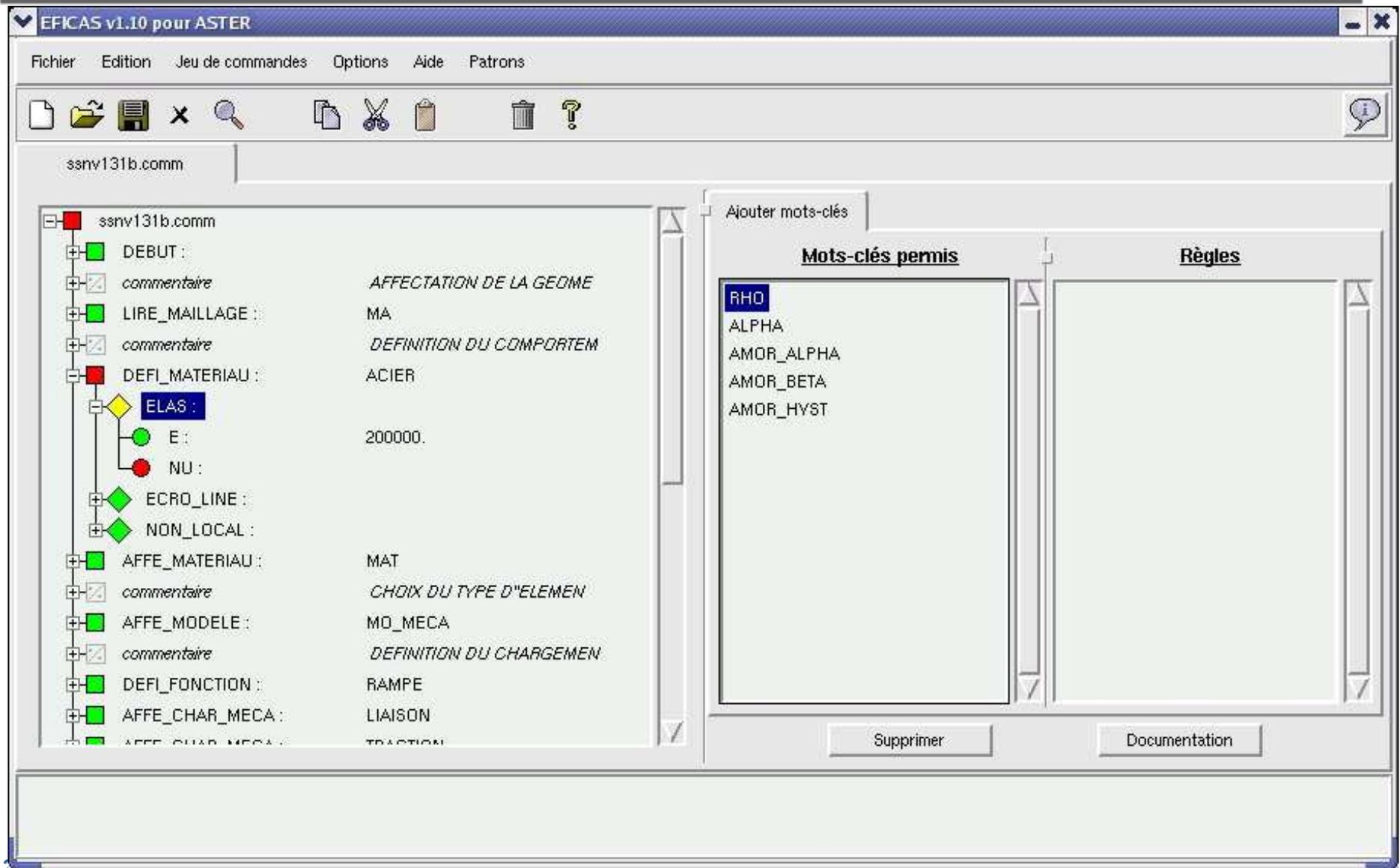
Choix d'un modèle de développement en interne sur base PYTHON

3 L'histoire du choix de python pour Aster

Arguments de choix de python et de l'architecture

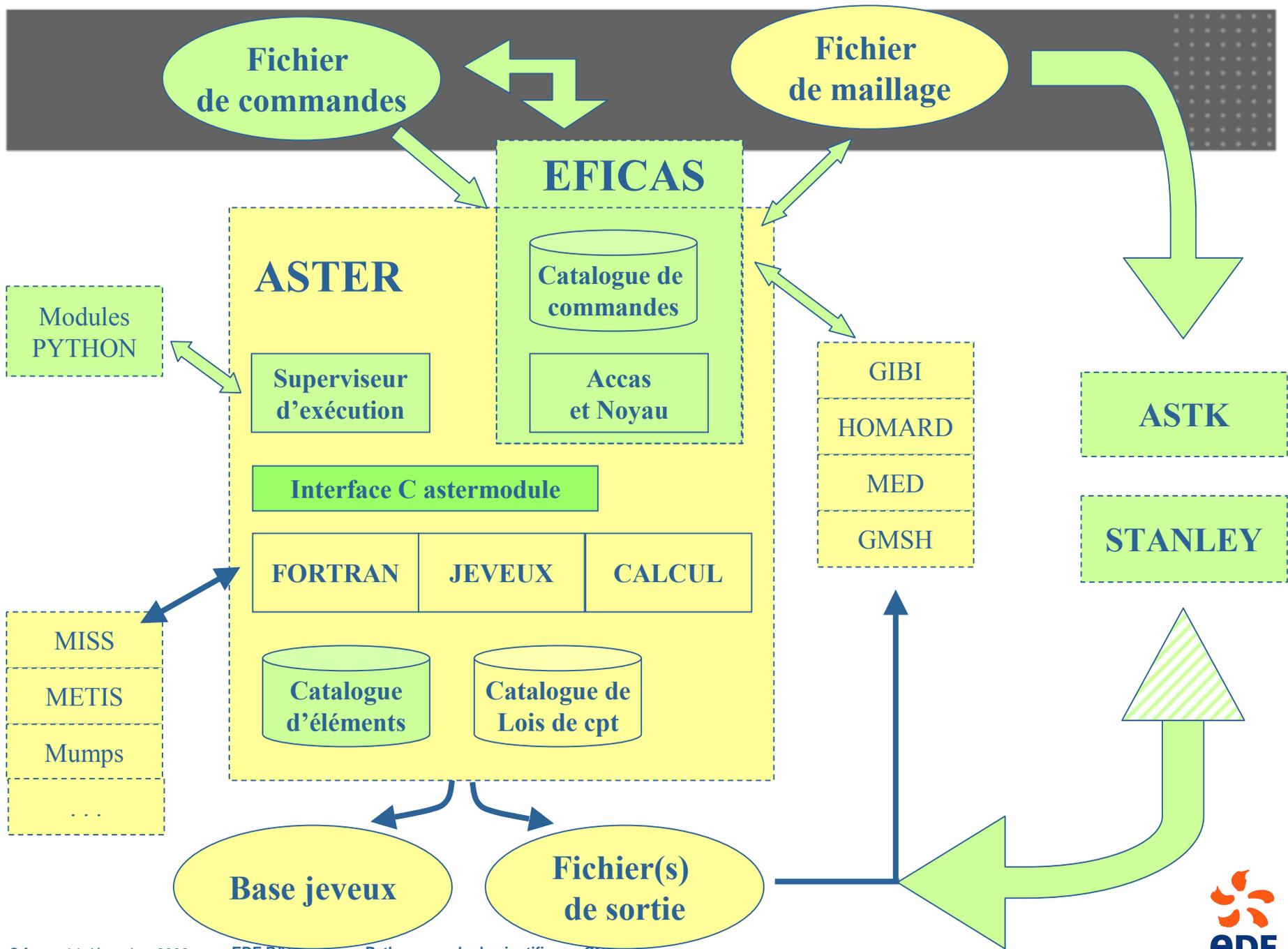
- **Langage existant plutôt que propriétaire : économie de l'analyseur syntaxique**
- **Acceptation par la communauté des développeurs**
- **Existence d'une base de programmes de même type**
- **Macro-commandes écrites dans le même langage que les jeux de données**
- **Partage de l'analyseur syntaxique avec l'éditeur de jeux de données**

3 L'assistant de mise en données : Eficas





Ce qui est python dans Aster Les opportunités offertes



4 Ce qui est python dans Aster

Cahier des charges 1999

- Fichiers de commandes
- Catalogue de commandes
- Analyseur syntaxique
- Superviseur d'exécution
- Macro-commandes
- Efficas

Opportunités apportées par python

- Concepts nativement en python : tables, formules
- Commandes (macros) de manipulation de ces concepts (impression, lecture, calcul)
- Macro-commandes métier, utilisateur, prototypes
- Gestionnaire d'études Astk
- Post-traitement graphique Stanley
- Catalogues de description des structures : éléments, {alarmes, SD, LdC}
- Mêler langage de commande et script de pilotage d'outil graphique (Salomé, Gmsh)
- Intergiciel Aster / Salomé

4 Ce que python a aussi permis

Architecture logicielle plus puissante, plus maintenable, plus élégante

- **Ré-ingéniering** du superviseur. Faculté d'expression des catalogues **plus riche**.
- Bénéfice de **modules python** : Numeric, Xmgrace, Tk, random ...
- Reprogrammation en python de familles de commandes pour lesquelles **Fortran était inadapté** (macros ; manipulations d'objets simples tables, fonctions, formules ; utilisation de structures python natives)

La distinction utilisateur / développeur s'estompe

- **Structures de contrôle** dans les jeux de données
- Passerelles python/fortran pour **manipulation des structures de données dans l'espace python** (macros, jeux de commandes)
- **Macros** pour applications métier, pour prototypes

4 Ce que python a aussi permis (suite)

L'environnement d'étude et les interfaces

- **Eficas** ... mais c'était dans le cahier des charges initial
- **Astk** : gestionnaire d'étude
- **Stanley** : assistant de post-traitement
- **Salomé** : intergiciel Aster dans Salomé, faculté d'importer des services Salomé dans les jeux de commandes et les macros.

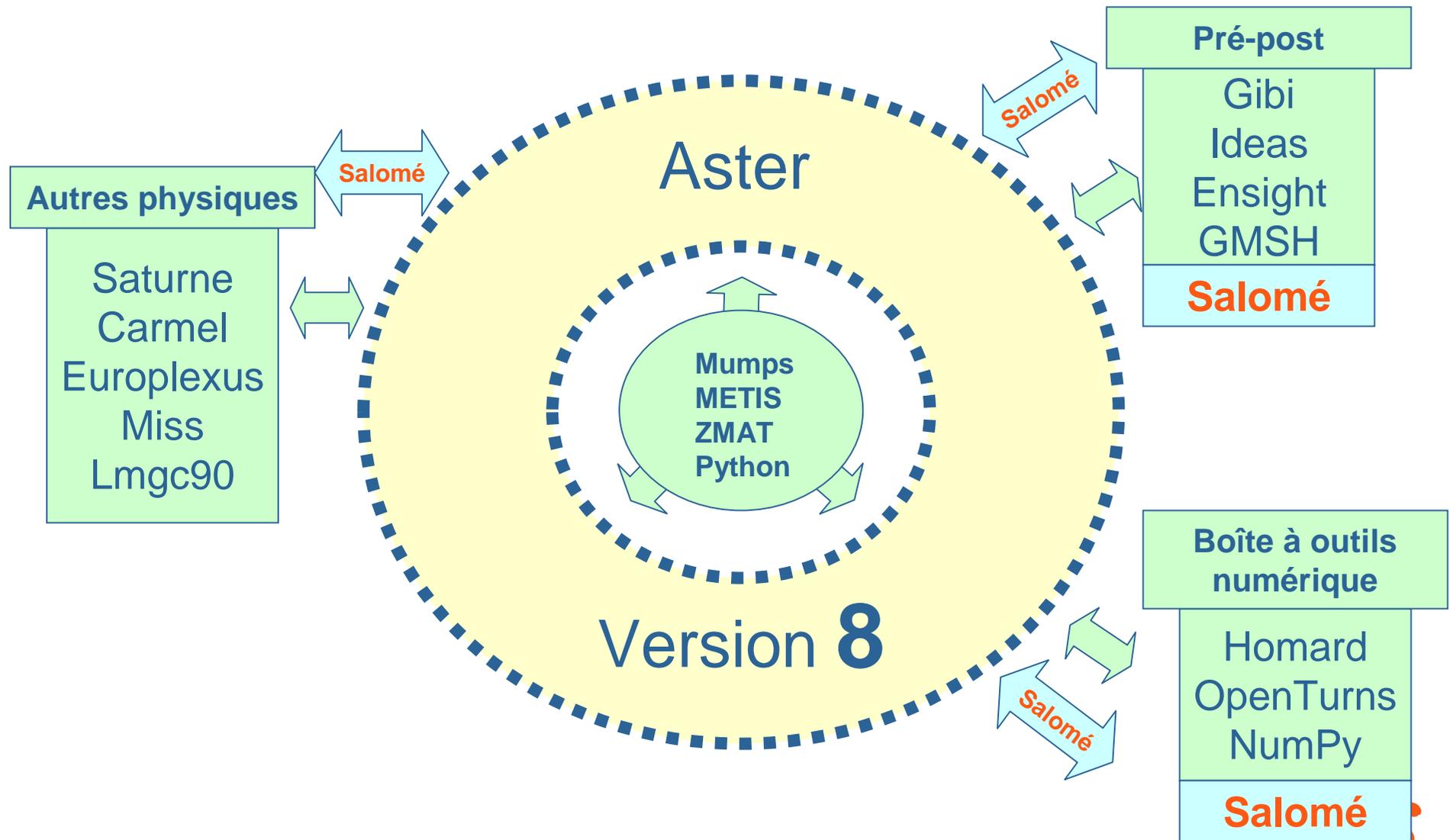
Ressources humaines et politique logicielle

- **Culture logiciel libre** : diffusion d'Aster-libre octobre 2001
- **Motivation des équipes** : surtout les jeunes
- **Nouvelle classe « intermédiaire » de développeur** qui n'intervient pas dans le fortran

4 Un logiciel peu ouvert : Aster version 5 (1998)



4 Un logiciel ouvert : Aster version 8 (2006)



4 Le vieux rêve à portée de clic : Aster-Salomé

The screenshot displays the SALOME 2.2.4 interface. The main window shows a 3D mesh of a mechanical part with a color-coded stress distribution. A vertical color bar on the left indicates the stress scale, ranging from 581 (blue) to 1.16e+03 (red). The interface includes a tree view on the left, a central 3D view, and a bottom panel with tabs for 'Fichier', 'Geometrie', 'Parametres', and 'Actions'. The 'Parametres' tab is active, showing a table of field components and geometric entities.

Champs	Composantes	Entites Geometriques	Ordres
EPSI_NOEU_DEPL	TOUT_CMP	TOUT_MALLAGE	TOUT_ORDRE
EQUI_ELGA_EPSI	VMIS	BloqBas (2D)	1
EQUI_ELNO_EPSI	TRESCA	BloqHaut (2D)	
EQUI_ELNO_SIGM	PRIN_1	Press (2D)	
EQUI_NOEU_EPSI	PRIN_2		
EQUI_NOEU_SIGM	PRIN_3		
ERRE_ELGA_NORE	VMIS_SG		
ERRE_ELNO_ELGA			
SIEF_ELGA_DEPL			
SIEF_ELNO_ELGA			

Invariants des contraintes aux noeuds



5

Demain Dangers et perspectives

5 Python, un danger pour le code ?

Risques AQ : la frontière utilisation / développement s'estompe

- Danger de perte de maîtrise des services aux utilisateurs (doc, hotline)
- Danger de fork par développements privés en python non restitués et plus facilement maintenables dans le temps que du fortran à compiler
- Danger sur la définition du périmètre « AQ » / « nonAQ » du code

Solution : définition stricte du périmètre AQ de l'usage, lobby auprès des « communautés » de développeurs

5 Python, un danger pour le code ? (suite)

Risques liés au langage

- Pérennité de python
- Usage de modules externes : risque de dépendance
- Une corrélation peut être plus forte qu'en fortran entre « qualité de programmation » et « performances »
- La communauté des développeurs Aster est composée de mécaniciens et de numériciens (non de purs informaticiens) : Fortran est plus accessible, rustique et maintenable. Il est plus facile de produire du code illisible en python.

Solution : formation des développeurs

5 Python, les performances ?

Une expérience de prototypage en python

- DEFI_PART_FETI : appel au partitionneur de maillage METIS en préparation au calcul // FETI. Création et manipulation de graphe à partir de la connectivité du maillage
- Prototypage rapide (2 semaines) : usage à profit des structures python dictionnaire, liste de liste de NumArray, dimensionnement dynamique
- Problème de performances : idem solution fortran pour 100.000 mailles ; rapport 50 pour 10^6 mailles
- Problème de duplication de la structure de données *maillage* dans l'espace python

Solution : (re) programmation Fortran

5 Python et Aster, l'avenir ?

Un jour, tout en python au dessus de *calcul* ?

calcul : routine d'encapsulation des calculs EF élémentaires

Structures de données dans la mémoire fortran, décrites et manipulées en python

Lien fort entre Aster et Salomé : environnement graphique, applications métiers

Aster module FEM dans Python ? [as Toolkit FEM Matlab]